Рубежный контроль №2
Студент: Попов М.Ю.
Группа: ИУ5-25М

Решение задачи классификации текстов.

Необходимо решить задачу классификации текстов на основе любого выбранного Вами датасета (кроме примера, который рассматривался в лекции). Классификация может быть бинарной или многоклассовой. Целевой признак из выбранного Вами датасета может иметь любой физический смысл, примером является задача анализа тональности текста.

Необходимо сформировать два варианта векторизации признаков - на основе CountVectorizer и на основе TfidfVectorizer.

В качестве классификаторов необходимо использовать два классификатора по варианту для Вашей группы:

ИУ5-25М, ИУ5И-25М, ИУ5И-26М SVC LogisticRegression

Импортируем библиотеки

```
from sklearn.feature_extraction.text import CountVectorizer,
TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import pandas as pd
import time
```

Загружаем набор данных

```
[2] # Загрузка данных
    ds = pd.read_csv('train_40k.csv')
    ds
```

	productId	Title	userId	Helpfulness	Score	Time	Text	Cat1	Cat2	Cat3
0	B000E46LYG	Golden Valley Natural Buffalo Jerky	A3MQDNGHDJU4MK	0/0	3.0	-1	The description and photo on this product need...	grocery gourmet food	meat poultry	jerky
1	B000GRA6N8	Westing Game	unknown	0/0	5.0	860630400	This was a great book!!!! It is well thought t...	toys games	games	unknown
2	B000GRA6N8	Westing Game	unknown	0/0	5.0	883008000	I am a first year teacher, teaching 5th grade....	toys games	games	unknown
3	B000GRA6N8	Westing Game	unknown	0/0	5.0	897696000	I got the book at my bookfair at school lookin...	toys games	games	unknown
4	B00000DMDQ	I SPY A is For Jigsaw Puzzle 63pc	unknown	2/4	5.0	911865600	Hi! I'm Martine Redman and I created this puzz...	toys games	puzzles	jigsaw puzzles
...	...	...	...	...	...	...	...	...	...	...
39995	B0006IYND6	Japonesque Silver Lipstick Palette Kit 1 piece	A1WKFQYYRJ95F6	0/0	3.0	1344211200	Even when it is very convenient I find it real...	beauty	makeup	lips
39996	B000A33FZY	Truform 20-30 Below Knee Closed-Toe, Beige, Small	A1OMZID10X2JY4	0/0	5.0	1344211200	Wore these the next day after eating Chinese t...	health personal care	medical supplies equipment	braces
39997	B000I7D2L4	Zadro Z300 Wall Mountable Fog Free Mirror	A3I8EYB4CKPQVO	0/0	3.0	1344211200	this is an OK product. Doesn't really stay "fo...	beauty	tools accessories	mirrors
39998	B000KHKKB2	Opalescent Glitter Lotion - 6.3 oz - Liquid	A3GTQQ5ZFCFBBL	0/0	4.0	1344211200	This "Glitter-Up" body lotion offers tremendou...	beauty	skin care	body
39999	B000JFC4C8	Pure Purple by Hugo Boss Eau De Parfum Spray 3...	A3JXBFTAV1EZET	0/0	1.0	1344211200	I was disappointed with the scent of this frag...	beauty	fragrance	women s

40000 rows × 10 columns

В качестве исходных данных был выбран набор данных `'train_40k.csv'`. В нём некоторые признаки содержат пропуски:

```
[3] # проверим пропуски в данных и устраним их
    na_mask = ds.isna()
    na_counts = na_mask.sum()
    na_counts
```

```
productId       0
Title          16
userId          0
Helpfulness     0
Score           0
Time            0
Text            0
Cat1            0
Cat2            0
Cat3            0
dtype: int64
```

```
ds.dropna(inplace=True)
na_mask = ds.isna()
na_counts = na_mask.sum()
na_counts
```

```
productId      0
Title          0
userId         0
Helpfulness    0
Score          0
Time           0
Text           0
Cat1           0
Cat2           0
Cat3           0
dtype: int64
```

Разделим набор данных на обучающую и тестувую выборки

```
X, Y = ds['Text'], ds['Cat1']
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2,
random_state=42)

time_arr = []
# векторизация признаков с помощью CountVectorizer
count_vect = CountVectorizer()
X_train_counts = count_vect.fit_transform(X_train)
X_test_counts = count_vect.transform(X_test)
# векторизация признаков с помощью TfidfVectorizer
tfidf_vect = TfidfVectorizer()
X_train_tfidf = tfidf_vect.fit_transform(X_train)
X_test_tfidf = tfidf_vect.transform(X_test)
```

Часть 1

Произведем обучения вдух классификаторов (по варианту) для CountVectorizer

```
# SVC
gbc = SVC()
start_time = time.time()
gbc.fit(X_train_counts, y_train)
train_time = time.time() - start_time
time_arr.append(train_time)
pred_gbc_counts = gbc.predict(X_test_counts)
print("Точность (CountVectorizer + SVC):", accuracy_score(y_test,
pred_gbc_counts))

# Logistic Regression
lr = LogisticRegression(max_iter=1000)
start_time = time.time()
lr.fit(X_train_counts, y_train)
train_time = time.time() - start_time
time_arr.append(train_time)
pred_lr_counts = lr.predict(X_test_counts)
print("Точность (CountVectorizer + LogisticRegression):",
accuracy_score(y_test, pred_lr_counts))
```

Часть 2

Произведем обучения вдух классификаторов (по варианту) для TfidfVectorizer

```
# SVC
gbc = SVC()
start_time = time.time()
gbc.fit(X_train_tfidf, y_train)
train_time = time.time() - start_time
time_arr.append(train_time)
pred_gbc_tfidf = gbc.predict(X_test_tfidf)
print("Точность (TfidfVectorizer + LinearSVC):", accuracy_score(y_test,
pred_gbc_tfidf))

# Logistic Regression
lr = LogisticRegression(max_iter=1000)
start_time = time.time()
lr.fit(X_train_tfidf, y_train)
train_time = time.time() - start_time
time_arr.append(train_time)
pred_lr_tfidf = lr.predict(X_test_tfidf)
print("Точность (TfidfVectorizer + LogisticRegression):",
accuracy_score(y_test, pred_lr_tfidf))
```

Сравним полученные результаты:

```
from tabulate import tabulate

data = [
    ["(CountVectorizer + LogisticRegression)", accuracy_score(y_test,
pred_lr_counts), time_arr[0]],
    ["(CountVectorizer + LinearSVC)", accuracy_score(y_test,
pred_gbc_counts), time_arr[1]],
    ["(TfidfVectorizer + LogisticRegression)", accuracy_score(y_test,
pred_lr_tfidf), time_arr[2]],
    ["(TfidfVectorizer + LinearSVC)", accuracy_score(y_test,
pred_gbc_tfidf), time_arr[3]]
]

sorted_data = sorted(data, key=lambda x: x[1], reverse=True)

print(tabulate(sorted_data, ['Модели','Точность валидации', 'Время
обучения'], tablefmt="grid"))
```

Модели	Точность валидации	Время обучения
(TfidfVectorizer + LinearSVC)	0.834063	30.3454
(TfidfVectorizer + LogisticRegression)	0.833938	917.7
(CountVectorizer + LogisticRegression)	0.817932	581.446
(CountVectorizer + LinearSVC)	0.755658	108.9

Вывод: Лучше всего показала себя пара TfidfVectorizer + LinearSVC